Explicit relation between all lower bound techniques for quantum query complexity*

Loïck Magnin[1,3] and Jérémie Roland[2,3]

1 **Centre for Quantum Technologies, National University of Singapore**
Block S15, 3 Science Drive 2, Singapore 117543
loick@locc.la
2 **QuIC, Ecole Polytechnique de Bruxelles, Université Libre de Bruxelles**
50 av. F.D. Roosevelt - CP165/59, B-1050 Bruxelles, Belgium
jroland@ulb.ac.be
3 **NEC Laboratories America**
4 Independence Way, Suite 200, Princeton NJ 08540, USA

Abstract

The polynomial method and the adversary method are the two main techniques to prove lower bounds on quantum query complexity, and they have so far been considered as unrelated approaches. Here, we show an explicit reduction from the polynomial method to the multiplicative adversary method. The proof goes by extending the polynomial method from Boolean functions to quantum state generation problems. In the process, the bound is even strengthened. We then show that this extended polynomial method is a special case of the multiplicative adversary method with an adversary matrix that is independent of the function. This new result therefore provides insight on the reason why in some cases the adversary method is stronger than the polynomial method. It also reveals a clear picture of the relation between the different lower bound techniques, as it implies that all known techniques reduce to the multiplicative adversary method.

1998 ACM Subject Classification F.1.1 Models of Computation

Keywords and phrases Quantum computation, lower bound, adversary method, polynomial method

Digital Object Identifier 10.4230/LIPIcs.STACS.2013.434

1 Introduction

Polynomial and adversary methods. There are two main techniques to prove lower bounds on quantum query complexity: the polynomial method [12, 20, 27], based on bounding the degree of the function seen as a polynomial, and adversary methods [15, 2, 11, 21, 19], based on bounding the change in a progress function from one query to the next. In its original form [2], the adversary method bounds the additive change in the progress function, hence we will call it *additive*, and the progress function is based on a matrix assigning positive weights to pairs of inputs. The polynomial method and this original adversary method are not comparable. Indeed, the original adversary method is limited by the "certificate complexity barrier" [30, 29], that is, for total functions, $\mathrm{ADV}(f) \leq \sqrt{C_0(f)C_1(f)}$ where $C_b(f)$ denotes

* This work was supported by ARO/NSA under grant W911NF-09-1-0569. L.M. also acknowledges the support of the Ministry of Education and the National Research Foundation, Singapore. J.R. also acknowledges support from the action *Mandats de Retour* of the *Politique Scientifique Fédérale Belge* and the Belgian ARC project COPHYMA.

30th Symposium on Theoretical Aspects of Computer Science (STACS'13).
Editors: Natacha Portier and Thomas Wilke; pp. 434–445

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

the certificate complexity of f for $f(x) = b$. It means that the original adversary method cannot prove lower bounds better than $\Omega(N^{1/2})$ for Element Distinctness. However, Aaronson and Shi [1] were able to prove a $\Omega(N^{2/3})$ lower bound using the polynomial method. On the other hand it is known that the adversary method can sometimes give better lower bounds than the polynomial method, in [4] Ambainis exhibits a function with polynomial degree d and adversary bound $\Omega(d^{1.3})$.

Høyer, Lee and Špalek have extended the additive adversary method by allowing negative weights in the matrix [18], and have shown that the corresponding bound, $\mathrm{ADV}^{\pm}(f)$, breaks the certificate complexity barrier. For simplicity, we will from now on refer to $\mathrm{ADV}^{\pm}(f)$ as the additive adversary bound, implicitly allowing negative weights.

Recently, a series of works [17, 7, 26, 25, 22] culminated in showing that this bound is tight in the bounded-error case for any function. However, this fundamental result does not answer all the questions about quantum query complexity as it suffers from two limitations. First, in some cases it is necessary to prove bounds for very small success probabilities, a regime where $\mathrm{ADV}^{\pm}(f)$ might not be tight. For this reason, while the optimality of the additive adversary bound implies that quantum query complexity satisfies a direct sum theorem, it cannot be used to prove a strong direct product theorem, which requires to prove nontrivial bounds for exponentially small success probabilities. Secondly, while the proof of optimality of $\mathrm{ADV}^{\pm}(f)$ implies that if a lower bound on the bounded-error quantum query complexity of a function can be proved with any method, it can also be proved with $\mathrm{ADV}^{\pm}(f)$, this reduction is not constructive. Concretely, there are still examples of lower bounds that can be proved using the polynomial method for which the optimal adversary matrix is unknown, a typical example being the Collision problem [1][1].

Multiplicative adversary method. The first limitation has been overcome thanks to the introduction of another adversary-type method. By formalizing an ad-hoc technique proposed by Ambainis, de Wolf and Špalek [5, 10], Špalek designed a new lower bound method which he called the multiplicative adversary method [28], as the idea is to bound the multiplicative change in the progress function for each query. Ambainis *et al.* [9] later showed that the multiplicative bound is always at least as strong as the additive one, and therefore also characterizes bounded-error quantum query complexity. Moreover, the multiplicative adversary method can prove better lower bounds for small success probability than the additive adversary method, and this was used to prove a strong direct product theorem for quantum query complexity [23].

Quantum state generation. Even when we are only interested in the quantum query complexity of functions, it is useful to also consider state generation problems: in that case, instead of producing the output $f(x)$ on input x, the algorithm is required to prepare a quantum state $|m_x\rangle$. Since unitary transformations independent of x may be applied without any query to x, a quantum state generation problem is completely defined by the Gram matrix $M = \sum_{x,x'} \langle m_{x'}|m_x\rangle |x\rangle\langle x'|$. In the special case of computing a function, M is a Boolean matrix. Thus every algorithm can be seen as generating a Gram matrix M. If the algorithm is allowed some error ε, then the set of Gram matrices that are acceptable outputs for the algorithm can be bounded by a so-called output condition. Different output conditions have been used before, for example, the original adversary method [2] was implicitly using a condition based on the L_∞ norm, while the adversary method with negative weights in [18]

[1] Until very recently it was also the case for the Element Distinctness problem, whose lower bound was proved by reduction to Collision, but a direct adversary lower bound has now been shown by Belovs [13], and later extended to the k−Sum problem by Belovs and Špalek [14].

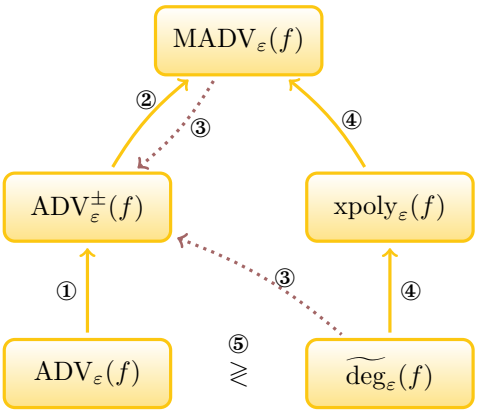

Figure 1 Relations between the different methods to prove lower bounds for quantum query complexity. An arrow from method A to method B implies that any lower bound that can be proved with A can also be proved with B (i.e., B is stronger than A). A solid yellow arrow means that the reduction is constructive, i.e., we can obtain a witness for B from a witness for A. ① [18] ② [9] ③ [25, 22] ④ [This article] ⑤ The original additive and the polynomial methods are incomparable [30, 29, 1, 4]

was implicitly using the factorization norm γ_2. Realizing that different output conditions could be combined with different (zero-error) lower bound methods was key to comparing the additive and multiplicative adversary methods in [9]. More recently, Lee and Roland [23] were able to characterize exactly the set of acceptable Gram matrices, hence providing an optimal output condition (see Claim 4), which allowed them to prove a strong direct product theorem for quantum query complexity. This also simplifies the study of lower bounds techniques as it implies that the bounded-error quantum query complexity of a problem can be studied by bounding the zero-error quantum query complexity of all Gram matrices that define valid output states for the problem. As a consequence it is sufficient to compare the zero-error bounds for two methods in order to compare them.

Our results. In this article, we tackle the second limitation by giving an explicit reduction from the polynomial method to the multiplicative adversary method. In order to do so, we introduce yet another lower bound technique for quantum query complexity, which we call the extended polynomial method (Definition 10 and Theorem 11) as it can be seen as an extension of the polynomial method to Gram matrices. As the degree of a Boolean function can be stated as the maximum index of its Fourier coefficients, that is, $\deg(f) = \max\{|S| : \langle \chi_S, f\rangle \neq 0\}$, we define the degree of a Gram matrix by the maximum index k such that the Gram matrix has support on a Fourier vector $|\chi_S\rangle$ with $|S| = k$, that is, $\deg(M) = \max\{|S| : \langle \chi_S|M|\chi_S\rangle \neq 0\}$.

For Boolean functions, the polynomial and the extended polynomial bounds are equal in the zero-error case. However, for the approximate case, the extended polynomial method uses the tight output condition, and is therefore possibly stronger than the polynomial method (Theorem 13).

We also compare the extended polynomial method to the multiplicative adversary method. More particularly, we show that in the limit $c \to \infty$, where c is the maximum multiplicative change in the progress function for one query, the multiplicative bound tends to the extended polynomial method (Theorem 14). This proof is constructive, i.e., we give an explicit multiplicative adversary matrix for which we have the equality. It might come as a surprise

that this matrix does not depend on the problem: it is the same adversary matrix for every function. Let us note that it was proved in [9] that the multiplicative bound is stronger than the additive bound in the limit $c \to 1$, that is, at the other end of the possible range for c. This new result therefore completes the picture of the relations between the different lower bound techniques in quantum query complexity (see Figure 1), and shows in particular that all these methods reduce to the multiplicative adversary method.

Many proofs are omitted from this extended abstract and can be found in the full version of the paper.

2 Preliminaries

2.1 Gram matrices and fidelity

▶ **Definition 1.** A **density matrix** ρ is a positive semidefinite matrix $\rho \succeq 0$ such that $\mathrm{tr}(\rho) = 1$. A **normalized Gram matrix** A is a positive semidefinite matrix $A \succeq 0$ such that $A \circ \mathbb{I} = \mathbb{I}$, where $\circ$ denotes the Hadamard (entry-wise) product.

Note that any positive semidefinite matrix A can be written as a Gram matrix in the broader sense, i.e., there always exists a set of vectors $\{|a_x\rangle\}$ such that $A_{xy} = \langle a_x|a_y\rangle$. Here, the additional constraint $A \circ \mathbb{I} = \mathbb{I}$ means that we require those vectors to have norm 1. Since all Gram matrices will be normalized in what follows, we will from now on refer to normalized Gram matrices as simply *Gram matrices*.

▶ **Definition 2.** The **fidelity** $\mathcal{F}(\rho, \sigma)$ between two density matrices ρ and σ is defined by $\mathcal{F}(\rho, \sigma) = \mathrm{tr}\sqrt{\sqrt{\rho}\,\sigma\sqrt{\rho}}$.

The **Hadamard product fidelity** $\mathcal{F}_H(A, B)$ between two Gram matrices A and B is defined by $\mathcal{F}_H(A, B) = \min_{|u\rangle:\||u\rangle\|=1} \mathcal{F}(A \circ |u\rangle\langle u|, B \circ |u\rangle\langle u|)$.

The notation $\mathcal{F}_H$ and the name Hadamard product fidelity[2] are new to this article, but this quantity has been proved to be the tight output condition for the quantum query complexity in [23] (see Claim 4 below).

2.2 Quantum query complexity

Consider a Boolean function $f : \{0,1\}^n \to \{0,1\}$. In the *black-box model*, we are interested in computing $f(x)$ when x is given by an oracle $O_x : |i, b\rangle \mapsto (-1)^{b \cdot x_i}|i, b\rangle$. We denote by $Q_\varepsilon(f)$ the quantum query complexity of f, i.e., the minimum number of queries to O_x necessary for any algorithm to output $f(x)$ with error at most ε (see, e.g., [16]). Note that our choice of oracle computes the bits of x in the phase. Another variant of this model considers an oracle that computes the bits in a register, but it can be shown that these models are equivalent.

Even when we are only interested in the quantum query complexity of functions, it is useful to also consider state generation problems [9, 22]. In that case, instead of producing the output $f(x)$ on input x, the algorithm is required to prepare a quantum state $|m_x\rangle \in \mathcal{H}$. Since unitary transformations independent of x may be applied without any query to x, a quantum state generation problem is completely defined by the Gram matrix $M = \sum_{x,x'} \langle m_{x'}|m_x\rangle |x\rangle\langle x'|$. For a quantum state generation problem specified by a Gram matrix M, we define two

[2] The name is chosen by analogy to the Hadamard product trace norm γ_2 (equivalent to the Hadamard product operator norm and also called factorization norm), which for Hermitian matrices can be written in the very similar form $\gamma_2(A) = \max_{|u\rangle:\||u\rangle\|\leq 1} \|A \circ |u\rangle\langle u|\|_{\mathrm{tr}}$.

different notions of query complexity. The coherent query complexity $Q_\varepsilon(M)$ is the minimum number of queries to the register oracle O_x necessary to generate a state $|n_x\rangle \in \mathcal{H} \otimes \mathcal{H}'$ such that $\Re(\langle n_x|(|m_x\rangle \otimes |\bar{0}\rangle)) \geq \sqrt{1-\varepsilon}$, where $\mathcal{H}'$ is the workspace of the algorithm, $|\bar{0}\rangle \in \mathcal{H}'$ is a default state for this workspace and $\Re(z)$ denotes the real part of a complex number z. The non-coherent query complexity $Q_\varepsilon^{\mathrm{nc}}(M)$ is defined similarly, except that it is enough to prepare a state $|n_x\rangle \in \mathcal{H} \otimes \mathcal{H}'$ such that $\Re(\langle n_x|(|m_x\rangle \otimes |m'_x\rangle)) \geq \sqrt{1-\varepsilon}$, for an arbitrary set of states $|m'_x\rangle \in \mathcal{H}'$ (that is, the workspace does not have to be reset to its default state).

For a Boolean function f, let us define the $\{1,-1\}$-valued function $\varphi : \{0,1\}^n \to \{1,-1\} : x \mapsto (-1)^{f(x)}$. There are two natural quantum state generation problems associated to f, corresponding to the Gram matrices $F = \sum_{x,x'} \delta_{f(x),f(x')}|x\rangle\langle x'|$ and $\Phi = \sum_{x,x'} \varphi(x)\varphi(x')|x'\rangle\langle x|$, where δ is the Kronecker delta. Indeed, generating the Gram matrix F non-coherently is exactly the same problem as computing f, and we therefore have $Q_\varepsilon(f) = Q_\varepsilon^{\mathrm{nc}}(F)$, while generating the Gram matrix Φ coherently corresponds to *computing the function in the phase*, i.e., we need to generate the state $\varphi(x)|\bar{0}\rangle$. The bounded-error complexities of these problems are closely related:

▶ **Claim 3** ([23]). $Q_{(1-\sqrt{1-\varepsilon})/2+\varepsilon/4}(f) \leq Q_\varepsilon(\Phi) \leq 2Q_{(1-\sqrt{1-\varepsilon})/2}(f)$.

This implies that to prove bounds on the bounded-error query complexity of f, it is sufficient to prove bounds on the query complexity of the related quantum state generation problem Φ, and this is precisely the approach that we will use in this article.

Another advantage of considering quantum state generation problems is that we can study the bounded-error query complexity of a problem by bounding the zero-error query complexity of all Gram matrices that define valid output states for the problem. It follows from the following claim that this set of valid Gram matrices is characterized by the Hadamard product fidelity:

▶ **Claim 4** ([23]). *For any Gram matrix M and any $\varepsilon \geq 0$, we have*

$$Q_\varepsilon(M) = \min_N \{Q_0(N) : \mathcal{F}_H(N,M) \geq \sqrt{1-\varepsilon},\ N \succeq 0,\ N \circ \mathbb{I} = \mathbb{I}\}.$$

2.3 The polynomial method

▶ **Definition 5.** For any $\varepsilon \geq 0$, the **approximate degree** $\widetilde{\deg}_\varepsilon(f)$ of a function $f : \{0,1\}^n \to \mathbb{R}$ is defined as $\widetilde{\deg}_\varepsilon(f) = \min_p \{\deg(p) : \forall x \in \{0,1\}^n,\ |p(x) - f(x)| \leq \varepsilon\}$, where the minimum is over n-variate polynomials $p : \mathbb{R}^n \to \mathbb{R}$.

▶ **Theorem 6** (Polynomial method [12]). *If f is a Boolean function, then* $Q_\varepsilon(f) \geq \Omega\Big(\widetilde{\deg}_\varepsilon(f)\Big)$.

In this article, we will use some basic Fourier analysis to relate degree of a function with Gram matrices. For the sake of readability, we will identify a set $S \subseteq \{1,\ldots,n\}$ with its characteristic vector $S \in \{0,1\}^n$: $S_i = 1$ if and only if $i \in S$, and thus $|S|$ can be either the cardinal of the set S or the Hamming weight of the vector S.

▶ **Definition 7.** For any $S \in \{0,1\}^n$, let us define $|\chi_S\rangle = \frac{1}{\sqrt{2^n}} \sum_x (-1)^{S \cdot x}|x\rangle$. For a function $\varphi : \{0,1\}^n \to \mathbb{R}$, define the (non-normalized) state $|\varphi\rangle = \frac{1}{\sqrt{2^n}} \sum_x \varphi(x)|x\rangle$. We define the S-th **Fourier coefficient** of φ as $\hat{\varphi}(S) = \langle \chi_S|\varphi\rangle$.

Let us note that the set $\{|\chi_S\rangle\}_S$ is an orthonormal basis and that by definition, we then have $\hat{\varphi}(S) = \frac{1}{2^n} \sum_x (-1)^{S.x}\varphi(x)$ and $\varphi(x) = \sum_S (-1)^{S.x}\hat{\varphi}(S)$, which are the usual Fourier transform over the hypercube and its inverse. With these notations, we can also write the degree of a function φ as $\deg(\varphi) = \max_S\{|S| : \hat{\varphi}(S) \neq 0\}$.

2.4 The multiplicative adversary method

Let us consider a quantum algorithm generating the Gram matrix M with error at most ε using T queries. Let $|\psi_x^t\rangle$ be the state of the algorithm right after the t-th query when the input is x, and $M^t = \sum_{x,x'} \langle\psi_{x'}^t|\psi_x^t\rangle |x\rangle\langle x'|$ be the corresponding Gram matrix. Note that $M^0 = \mathbb{J}$ and $M^T \approx M$ (more precisely $\mathcal{F}_H(M^T, M) \geq \sqrt{1-\varepsilon}$). The basic idea of all adversary methods is to design a Hermitian matrix W defining a progress function $W[M] = \mathrm{tr}[WM]$ such that the initial value $W[\mathbb{J}]$ is low and the final value $W[M^T]$ is high (or vice versa), and then to bound the maximal change in the progress function for any oracle call. Whereas the additive method bounds the difference $|W[M^{t+1}] - W[M^t]|$, the multiplicative method bounds the ratio $W[M^{t+1}]/W[M^t]$. In this paper we use the definition of the multiplicative adversary method given by [23] which is a slight extension of the original multiplicative adversary method in [28].

▶ **Definition 8.** Let M be a Gram matrix specifying a quantum state generation problem and for all $i \in \{1, \cdots, n\}$, $D_i = \sum_{x,x'} (-1)^{x_i + x'_i} |x\rangle\langle x'|$ the action of the phase oracle on input i. Fix $c > 1$. The **multiplicative adversary bounds** are:

$$\mathrm{MADV}_0^c(M) = \frac{1}{\log c} \max_{W \succeq 0} \left\{ \log \mathrm{tr}[WM] : \mathrm{tr}[W\mathbb{J}] = 1,\ W \circ D_i \preceq cW\ \forall i \right\},$$
$$\mathrm{MADV}_\varepsilon^c(M) = \min_N \left\{ \mathrm{MADV}_0^c(N) : \mathcal{F}_H(N, M) \geq \sqrt{1-\varepsilon},\ N \succeq 0,\ N \circ \mathbb{I} = \mathbb{I} \right\},$$
$$\mathrm{MADV}_\varepsilon(M) = \sup_{c>1} \mathrm{MADV}_\varepsilon^c(M).$$

We call **adversary matrix** for $\mathrm{MADV}_0^c(M)$ any matrix $W \succeq 0$ such that $\mathrm{tr}[W\mathbb{J}] = 1$ and $W \circ D_i \preceq cW$ for all i.

▶ Remark. Let us note that the parameter c represents the maximum multiplicative change in the progress function that can result from one query. Since, for any matrix $W \succ 0$, the constraint $W \circ D_i \preceq cW$ is always satisfied for $c \geq \left\| (W \circ D_i)^{1/2} W^{-1/2} \right\|^2$, one could directly obtain the multiplicative bound MADV_0 by optimizing over W and taking $c = \left\| (W \circ D_i)^{1/2} W^{-1/2} \right\|^2$. However, it is useful to define the bound MADV_0^c for fixed c as this can be expressed as a semidefinite program (see [23]), where the objective value is optimized over W. The best bound on the quantum query complexity is then obtained by maximizing the objective value over both W and c.

▶ **Theorem 9** (Multiplicative adversary [28, 23])**.** *For any $\varepsilon \geq 0$ and any Gram matrix M, we have $Q_\varepsilon(M) \geq \mathrm{MADV}_\varepsilon(M)$.*

3 The extended polynomial method

We now extend the polynomial method from Boolean functions to Gram matrices.

▶ **Definition 10.** Let M be a Gram matrix specifying a quantum state generation problem. The **extended polynomial bounds** are

$$\mathrm{xpoly}_0(M) = \max_S \{|S| : \mathrm{tr}\left[|\chi_S\rangle\langle\chi_S| M\right] \neq 0\},$$
$$\mathrm{xpoly}_\varepsilon(M) = \min_N \left\{ \mathrm{xpoly}_0(N) : \mathcal{F}_H(N, M) \geq \sqrt{1-\varepsilon},\ N \succeq 0,\ N \circ \mathbb{I} = \mathbb{I} \right\}.$$

▶ **Theorem 11.** *For any $\varepsilon \geq 0$ and any Gram matrix M, we have $Q_\varepsilon(M) \geq \mathrm{xpoly}_\varepsilon(M)$.*

Proof. We prove the statement for $\varepsilon = 0$ and the general case immediately follows from Claim 4 and the definition of $\mathrm{xpoly}_\varepsilon(M)$. This proof actually considers the extended polynomial method as an adversary method. Let us define the progress function

$$W[M^t] = \max_S \left\{ |S| : \mathrm{tr}[|\chi_S\rangle\langle\chi_S| M^t] \neq 0 \right\}.$$

Since $M^0 = \mathbb{J} = 2^n |\chi_\emptyset\rangle\langle\chi_\emptyset|$, its initial value is $W[M^0] = 0$. The final value is $W[M^T] = \mathrm{xpoly}_0(M)$. It suffices to show that one query increases the progress function by at most one.

Let $M^t = \sum_i M_i^t$ be the Gram matrix just before the $(t+1)$-th query, where M_i^t is the reduced Gram matrix corresponding to the part of the state where bit x_i is queried (see, e.g., [9] for details). Let $k = W[M^t]$ and note that by positivity, we have $\mathrm{tr}[|\chi_S\rangle\langle\chi_S| M^t] = 0$ if and only if $\mathrm{tr}[|\chi_S\rangle\langle\chi_S| M_i^t] = 0$ for all i. Therefore, we also have $W[M_i^t] \leq k$ for any i.

After the query, the Gram matrix of the algorithm will be $M^{t+1} = \sum_i M_i^t \circ D_i$. Let us observe that for any matrix A, we have $A \circ D_i = U_i A U_i^\dagger$ where $U_i = U_i^\dagger$ is the unitary matrix $U_i = \sum_x (-1)^{x_i} |x\rangle\langle x|$. In particular, $|\chi_S\rangle\langle\chi_S| \circ D_i = |\chi_{S'}\rangle\langle\chi_{S'}|$ where $S' = S \cup \{i\}$ if $i \notin S$ and $S' = S \setminus \{i\}$ if $i \in S$.

For all $S \in \{0,1\}^n$, we get:

$$\mathrm{tr}\left[|\chi_S\rangle\langle\chi_S|(M_i^t \circ D_i)\right] = \mathrm{tr}\left[(|\chi_S\rangle\langle\chi_S| \circ D_i) M_i^t\right] = \sum_{T:|T|\leq k} \mathrm{tr}\left[(|\chi_S\rangle\langle\chi_S| \circ D_i)|\chi_T\rangle\langle\chi_T| M_i^t\right].$$

This quantity is null for all S such that $|S| > k+1$, therefore the progress function can increase by at most one per query. ◀

We have defined the extended polynomial method with the Fourier basis, but one might wonder if choosing another basis could provide better bounds. It turns out that this is not the case.

▶ **Claim 12.** *Let $\{\Pi_k : 0 \leq k \leq K\}$ be a set of orthogonal projectors such that*

1. $\sum_k \Pi_k = \mathbb{I}_{\mathbb{C}^{2^n}}$,
2. $\mathrm{tr}(\Pi_0 \mathbb{J}) = 2^n$,
3. $\forall i \in \{1,\ldots,n\},\ \forall l,k$ *such that* $|l-k| > 1,\ \mathrm{tr}[(\Pi_l \circ D_i)\Pi_k] = 0$.

Then, for any Gram matrix M, we have $Q_0(M) \geq \mathrm{xpoly}_0(M) \geq \max_k \{k : \mathrm{tr}(\Pi_k M) \neq 0\}$.

Therefore, while any set of projectors provides a lower bound on quantum query complexity, the best bound is achieved by the extended polynomial method, which corresponds to the special case $K = n$ and $\Pi_k = \sum_{S:|S|=k} |\chi_S\rangle\langle\chi_S|$.

4 Relation between the polynomial and the extended polynomial methods

In this section, we compare the strength of the polynomial and the extended polynomial methods. Let f be a Boolean function and Φ the Gram matrix corresponding to computing f in the phase. By definition of the extended polynomial method, we have that $\mathrm{xpoly}_0(\Phi) = \deg(f)$. However the equality is lost in the approximate case:

▶ **Theorem 13.** *Let f be a Boolean function and Φ be the Gram matrix corresponding to computing f in the phase. For any $\varepsilon \geq 0$, we have $\mathrm{xpoly}_\varepsilon(\Phi) \geq \widetilde{\deg}_{\varepsilon/2}(f)$.*

Proof (sketch). Let N be a Gram matrix achieving the minimum in the definition of $\mathrm{xpoly}_\varepsilon(\Phi)$, that is, an optimal final Gram matrix of an algorithm for Φ. We first express this Gram matrix as $N = \sum_{x,y} \langle\psi_x|\psi_y\rangle |y\rangle\langle x|$, where $|\psi_x\rangle = \sum_i p_i(x)|i\rangle$ is the final state of the algorithm on input x expressed in the computational basis. By definition of the extended polynomial bound, we then have $\mathrm{xpoly}_\varepsilon(\Phi) = \max_i(\deg(p_i))$, where the maximum is over polynomials satisfying the normalization constraint $\sum_i p_i(x)^2 = 1$ and the correctness constraint $(-1)^{f(x)}\Re(p_0(x)) \geq \sqrt{1-\varepsilon}$, for all inputs x. The polynomial p_0 then witnesses that $\widetilde{\deg}_{\varepsilon/2}(f) \leq \deg(p_0) \leq \mathrm{xpoly}_0(N)$. ◀

5 Relation with the multiplicative adversary method

In [9], it was shown that in the limit $c \to 1$, the multiplicative adversary bound $\mathrm{MADV}_0^c(M)$ is at least as strong as the additive adversary bound $\mathrm{ADV}^{\pm}(M)$. Here, we show that the extended polynomial bound can be obtained by taking the limit $c \to \infty$.

▶ **Theorem 14.** *Let M be a Gram matrix, $\varepsilon \geq 0$, $T = \mathrm{xpoly}_\varepsilon(M)$ and $\Pi_{\geq T} = \sum_{S:|S|\geq T} |\chi_S\rangle\langle\chi_S|$. Moreover, let $\delta > 0$ be such that $\mathrm{tr}[\Pi_{\geq T} N] \geq \delta$ for any Gram matrix N such that $\mathcal{F}_H(N, M) \geq \sqrt{1-\varepsilon}$. Then, for any $c > 1$, we have*

$$\mathrm{xpoly}_\varepsilon(M) - \frac{n - \log\delta}{\log c} \leq \mathrm{MADV}_\varepsilon^c(M) \leq \mathrm{xpoly}_\varepsilon(M) + \frac{n}{\log c}.$$

In particular, in the limit $c \to \infty$, we have $\lim_{c\to\infty} \mathrm{MADV}_\varepsilon^c(M) = \mathrm{xpoly}_\varepsilon(M)$.

▶ Remark. Note that such a value of δ always exists. Assume by contradiction that $\mathrm{tr}[\Pi_{\geq T} N] = 0$, then $\mathrm{xpoly}_0(N) \leq T - 1$, however N is an ε-approximation of M that has a polynomial bound of T.

The general idea of the proof is to consider the multiplicative adversary matrix

$$W = \frac{1}{2^n} \sum_S c^{|S|} |\chi_S\rangle\langle\chi_S|$$

as a multiplicative adversary matrix. The lower bound then follows from the fact that in the limit $c \to \infty$, the value of the progress function $W[M] = \mathrm{tr}[WM]$ will be dominated by the term in $c^{|S|}$ for the set S with the largest size $|S| = k$ such that $\langle\chi_S|M|\chi_S\rangle \neq 0$, which therefore corresponds to the degree of the matrix M. As for the upper bound, we show that the matrix W becomes an optimal multiplicative adversary matrix in the limit $c \to \infty$. This can be shown by observing that one oracle call can only map a Fourier basis state $|\chi_S\rangle$ to another Fourier basis state $|\chi_{S'}\rangle$ with $|S'| = |S| \pm 1$ which implies bounds on the elements of any possible multiplicative adversary matrix written in the Fourier basis.

Proof. We prove it for the zero-error case, the general case follows immediately.

Consider the matrix $W = \frac{1}{2^n} \sum_S c^{|S|} |\chi_S\rangle\langle\chi_S|$. It is a valid adversary matrix for $\mathrm{MADV}_0^c(M)$ since $\mathrm{tr}[W\mathbb{J}] = 1$ and $W \circ D_i \preceq cW$, $\forall i \in \{1, \ldots, n\}$. This inequality follows from $W \circ D_i = \frac{1}{2^n}\left(\sum_{S:i\in S} c^{|S|-1}|\chi_S\rangle\langle\chi_S| + \sum_{S:i\notin S} c^{|S|+1}|\chi_S\rangle\langle\chi_S|\right)$, see proof of Theorem 11. Let W' be an optimal multiplicative adversary matrix for $\mathrm{MADV}_0^c(M)$. Let us show that $\mathrm{tr}[WM] \leq \mathrm{tr}[W'M] \leq 2^n\,\mathrm{tr}[WM]$.

The first inequality is a direct consequence of the fact that W is an adversary matrix for $\mathrm{MADV}_0^c(M)$ and the definition of the multiplicative adversary bound.

To prove the second inequality, let us first show by induction on $k = |S|$ that $\langle\chi_S|W'|\chi_S\rangle \leq \frac{1}{2^n} c^{|S|}$ for any set S. For $k = 0$, the condition $\mathrm{tr}[W'\mathbb{J}] = 1$ is equivalent to $\langle\chi_\emptyset|W'|\chi_\emptyset\rangle = \frac{1}{2^n}$.

Let us fix $0 \leq k \leq n$, and assume that $\forall S$ such that $|S| = k$, we have $\langle \chi_S | W' | \chi_S \rangle \leq \frac{1}{2^n} c^k$. Let S' be a set of size $k+1$ and decompose it into $S' = S \cup \{i\}$. Observe first that $\langle \chi_S | W' \circ D_i | \chi_S \rangle = \langle \chi_S | U_i W' U_i | \chi_S \rangle = \langle \chi_{S'} | W' | \chi_{S'} \rangle$ where $U_i = \sum_x (-1)^{x_i} |x\rangle\langle x|$ as defined in the proof of Theorem 11. Hence by sandwiching $W' \circ D_i \preceq cW'$ with $|\chi_S\rangle$, we get $\langle \chi_{S'} | W' | \chi_{S'} \rangle \leq c \langle \chi_S | W' | \chi_S \rangle \leq \frac{1}{2^n} c^{|S|+1}$.

We can now proceed with the rest of the proof:

$$\begin{aligned} \operatorname{tr}[W'M] &= \sum_S \langle \chi_S | W'M | \chi_S \rangle = \sum_{S,S'} \langle \chi_S | W' | \chi_{S'} \rangle \langle \chi_{S'} | M | \chi_S \rangle \\ &\leq \sum_{S,S'} |\langle \chi_S | W' | \chi_{S'} \rangle| \, |\langle \chi_{S'} | M | \chi_S \rangle| \,. \end{aligned}$$

We now use the property that for any positive semidefinite matrix A, $|A_{ij}| \leq \sqrt{A_{ii} A_{jj}}$,

$$\operatorname{tr}[W'M] \leq \left(\sum_S \sqrt{\langle \chi_S | W' | \chi_S \rangle \langle \chi_S | M | \chi_S \rangle} \right)^2 .$$

Using the Cauchy-Schwarz inequality, we get:

$$\operatorname{tr}[W'M] \leq 2^n \sum_S \langle \chi_S | W' | \chi_S \rangle \langle \chi_S | M | \chi_S \rangle \leq \sum_S c^{|S|} \langle \chi_S | M | \chi_S \rangle = 2^n \operatorname{tr}[WM].$$

We are now ready to conclude the proof. From $\operatorname{tr}[WM] \leq \operatorname{tr}[W'M] \leq 2^n \operatorname{tr}[WM]$, we have by definition of $\mathrm{MADV}_0^c(M)$

$$\frac{\log \operatorname{tr}[WM]}{\log c} \leq \mathrm{MADV}_0^c(M) \leq \frac{n + \log \operatorname{tr}[WM]}{\log c}.$$

For $T = \mathrm{xpoly}_\varepsilon(M)$, we find from the first inequality

$$\mathrm{MADV}_0^c(M) \geq \frac{\log \frac{1}{2^n} c^T \operatorname{tr}[\Pi_{\geq T} M]}{\log c} = T + \frac{\log(\operatorname{tr}[\Pi_{\geq T} M]) - n}{\log c}.$$

Similarly, from the second inequality, we have

$$\mathrm{MADV}_0^c(M) \leq \frac{\log \sum_S c^{|S|} \langle \chi_S | M | \chi_S \rangle}{\log c} \leq T + \frac{\log \sum_S \langle \chi_S | M | \chi_S \rangle}{\log c} = T + \frac{n}{\log c},$$

where we used the facts that $\langle \chi_S | M | \chi_S \rangle = 0$ whenever $|S| > T$, and $\sum_S \langle \chi_S | M | \chi_S \rangle = \operatorname{tr}[M] = 2^n$. ◀

We note that $\mathrm{MADV}_\varepsilon^c(M)$ approaches its limiting value $\mathrm{xpoly}_\varepsilon(M)$ if c is large enough compared to $2^n/\delta$. In general, we cannot give a lower bound on δ in order to determine how large c should be. However, for the special case of Boolean functions, and comparing to the standard polynomial method, i.e., the approximate degree $\widetilde{\deg}_\varepsilon(f)$, instead of $\mathrm{xpoly}_\varepsilon(M)$, we can show that $\mathrm{MADV}_\varepsilon^c(\Phi)$ becomes at least as strong as $\widetilde{\deg}_\varepsilon(f)$ as soon as c is large compared to $2^n/\varepsilon$.

▶ **Lemma 15.** *Let f be a Boolean function with associated phase matrix Φ. Then, for any $c > 1$, we have* $\mathrm{MADV}_\varepsilon^c(\Phi) \geq \widetilde{\deg}_\varepsilon(f) - 2 \cdot \frac{n - \log \varepsilon}{\log c}$.

Proof. Just as in the proof of Theorem 13, we express the Gram matrix achieving the minimum in the definition of $\mathrm{MADV}_\varepsilon^c(\Phi)$ as $N = \sum_{x,y} \langle \psi_x | \psi_y \rangle |y\rangle\langle x|$, where $|\psi_x\rangle = \sum_i p_i(x) |i\rangle$ is the final state of the algorithm on input x expressed in the computational basis. After

relaxing the normalization condition on the states $|\psi_x\rangle$, we obtain that $\mathrm{MADV}^c_\varepsilon(\Phi) \geq \frac{1}{\log c} \log \frac{1}{2^n} \sum_S c^{|S|} \left|\hat{p}(S)\right|^2$, where p is the minimum taken over all polynomials $q : \{0,1\}^n \mapsto \mathbb{R}$ satisfying $\sqrt{1-\varepsilon} \leq (-1)^{f(x)} q(x) \leq 1$ for any $x \in \{0,1\}^n$.

By definition of $\mathrm{MADV}^c_\varepsilon(\Phi)$, we can then show that similarly to the lower bound in Theorem 14, we have $\mathrm{MADV}^c_\varepsilon(\Phi) \geq T - \frac{n - \log \delta}{\log c}$, where in this case $T = \widetilde{\deg}_\varepsilon(f)$ and $\delta = \sum_{S:|S|\geq T} \left|\hat{p}(S)\right|^2$. It can then be shown that δ must be at least $\frac{\varepsilon^2}{2^n}$, otherwise truncating the high Fourier coefficients from p would yield a polynomial witnessing that $\widetilde{\deg}_\varepsilon(f) < T$, a contradiction. ◀

Note that a similar argument cannot be used for the extended polynomial method because truncating the large Fourier coefficients from a Gram matrix N might yield a matrix that is not normalized (i.e., violating the constraint $N \circ \mathbb{I} = \mathbb{I}$).

6 Discussion and open questions

Strong connections have been known for quite some time between the approximate degree of a function and its query complexity: they are polynomially related for all (total) functions for classical complexity [24] as well as for quantum complexity [12]. The latter is actually often equal to the approximate degree (at least up to a constant factor) for many functions, including all symmetric functions and random functions. With a large number of tight bounds proved using the polynomial method [12, 1, 3, 8] to cite only a few, this method might even seem ubiquitous. However, it is not always tight as in some rare cases the adversary method is known to yield better bounds. By clarifying the relation between the polynomial and adversary bounds, this work provides some new insight on why this can be the case.

First, we showed that the polynomial method is a relaxation of a more general method which we called the extended polynomial method. This has a particularly nice interpretation when one wants to compute the value of a function in a register, i.e., the goal is to prepare the state $|f(x)\rangle$.[3] When error ε is allowed, measuring this register should yield outcome $f(x)$ with probability at least $1-\varepsilon$, that is, the probability $p(x)$ of obtaining outcome 1 should be close to 1 when $f(x) = 1$ and close to 0 when $f(x) = 0$. While the polynomial method only considers the degree of the probability $p(x)$, the extended polynomial method considers the degree of all the amplitudes in the final state of the algorithm, including the erroneous part. In terms of Gram matrices this corresponds to relaxing the condition $N \circ \mathbb{I} = \mathbb{I}$ to $N \circ \mathbb{I} \preceq \mathbb{I}$.[4]

In general it is not known how large the gap between the polynomial and the extended polynomial method can be. It appears to be larger by at least a factor two for some functions. Indeed, Ambainis *et al.* improved the lower bound for random Boolean functions from $n/4 - o(n)$ using the polynomial method, to $n/2 - o(n)$ (which is tight) by bounding the degree of all amplitudes in the final state of the algorithm [6] (their argument can be seen as a special case of the extended polynomial method).

[3] This is the standard problem studied in most articles on quantum query complexity, even though some recent works including this one have considered the problem of computing the function in the phase. Recall that Claim 4 implies that both problems are equivalent.

[4] Note that with the relaxed condition $N \circ \mathbb{I} \preceq \mathbb{I}$, the matrix N does not have to be a *normalized* Gram matrix anymore, in which case the Hadamard product fidelity is not defined. However, one can use another output condition, for example $\gamma_2(N - M) \leq \sqrt{2\varepsilon}$, where γ_2 denotes the Hadamard product trace norm. These output conditions are related up to a constant [22, 23], so that it only affects the lower bound by at most a constant factor for bounded-error query complexity.

Secondly this provides a partial answer on how the multiplicative adversary method MADV^c varies with c. Indeed, while it was already known that $\mathrm{MADV}_\varepsilon^{c\to 1}(f) \geq \mathrm{ADV}_\varepsilon^{\pm}(f)$, we have proved that $\mathrm{MADV}_\varepsilon^{c\to\infty}(f) \geq \widetilde{\deg}_\varepsilon(f)$, and in particular, $\mathrm{MADV}_0^{c\to\infty}(f) = \deg(f)$ in the zero-error case. This implies that the gap between MADV and $\mathrm{MADV}^{c\to\infty}$ can be at least polynomially large by considering the Ambainis function [4], for which the polynomial method fails to give a tight bound, contrary to the adversary method. This gap might be explained by the fact that in the limit $c \to \infty$, the eigenbasis of the best adversary matrix is restricted to be the Fourier basis, while for smaller values, other bases can provide better bounds.

To summarize our current knowledge, the situation is the following. On the one hand, when c tends to one, the multiplicative adversary method is tight for bounded-error ([9]) but not for zero-error (e.g., for the OR function, there is a quadratic gap). On the other hand, when c tends to infinity, the multiplicative method seems better for zero-error as it proves the $\Omega(n)$ lower bound for OR, but it is not always tight (Ambainis function). As for low success probability, it seems that taking c bounded away from one provides an advantage, as shown in particular by the strong direct product theorems proved using the multiplicative [28, 23] and polynomial methods [20, 27].

This leaves open a few interesting questions about the behavior of the multiplicative adversary method. Can we say more about the dependence of MADV^c on c? Can we improve the relation $\mathrm{MADV}_\varepsilon^{c\to 1}(M) \geq \mathrm{ADV}_\varepsilon^{\pm}(M)$ to an equality in general? Can we characterize the set of functions for which the (extended or not) polynomial method does not provide a tight bound? Finally, does the multiplicative adversary method characterize the quantum query complexity, i.e., is it tight for any error?

Acknowledgements Most of this work was done at NEC Laboratories America. The authors thank M. Rötteler, D. Gavinsky, and T. Lee for stimulating discussions; and R. de Wolf and R. Špalek for interesting comments. They also thank R. Špalek for proposing the alternative proof of Lemma 15 using dual polynomials.

References

1 S. Aaronson and Y. Shi. Quantum lower bounds for the collision and the element distinctness problems. *J. ACM*, 51(4):595–605, 2004.

2 A. Ambainis. Quantum lower bounds by quantum arguments. *J. Comput. Sys. Sci.*, 64(4):750–767, 2002.

3 A. Ambainis. Polynomial degree and lower bounds in quantum complexity: Collision and element distinctness with small range. *Theor. Comput.*, 1:37–46, 2005.

4 A. Ambainis. Polynomial degree vs. quantum query complexity. *J. Comput. Sys. Sci.*, 72(2):220–238, 2006.

5 A. Ambainis. A new quantum lower bound method, with an application to strong direct product theorem for quantum search. *Theor. Comput.*, 6:1–25, 2010.

6 A. Ambainis, A. Bačkurs, J. Smotrovs, and R. de Wolf. Optimal quantum query bounds for almost all Boolean functions. In *Proc. STACS'13*, 2013.

7 A. Ambainis, A. M. Childs, B. W. Reichardt, R. Špalek, and S. Zhang. Any AND-OR formula of size N can be evaluated in time $N^{1/2+o(1)}$ on a quantum computer. *SIAM J. Comput.*, 39(6):2513–2530, 2010.

8 A. Ambainis and R. de Wolf. How low can approximate degree and quantum query complexity be for total boolean functions? *arXiv:1206.0717*, 2012.

9 A. Ambainis, L. Magnin, M. Roetteler, and J. Roland. Symmetry-assisted adversaries for quantum state generation. In *Proc. CCC'11*, pages 167–177, 2011.

10 A. Ambainis, R. Špalek, and R. de Wolf. A new quantum lower bound method, with applications to direct product theorems and time-space tradeoffs. In *Proc. STOC'06*, pages 618–633, 2006.

11 H. Barnum and M. Saks. A lower bound on the quantum query complexity of read-once functions. *J. Comput. Sys. Sci.*, 69(2):244–258, 2004.

12 R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf. Quantum lower bounds by polynomials. *J. ACM*, 48:778–797, 2001.

13 A. Belovs. Adversary lower bound for element distinctness. *arXiv:1204.5074*, 2012.

14 A. Belovs and R. Špalek. Adversary lower bound for the k-sum problem. In *Proc. ITCS'13*, 2013.

15 C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM J. Comput.*, 26(5):1510–1523, 1997.

16 H. Buhrman and R. de Wolf. Complexity measures and decision tree complexity: A survey. *Theor. Comput. Sci.*, 288(1):21–43, 2002.

17 E. Farhi, J. Goldstone, and S. Gutmann. A quantum algorithm for the Hamiltonian NAND tree. *Theor. Comput.*, 4:169–190, 2008.

18 P. Høyer, T. Lee, and R. Špalek. Negative weights make adversaries stronger. In *Proc. STOC'07*, pages 526–535, 2007.

19 P. Høyer, J. Neerbek, and Y. Shi. Quantum complexities of ordered searching, sorting, and element distinctness. *Algorithmica*, 34(4):429–448, 2008.

20 H. Klauck, R. Špalek, and R. de Wolf. Quantum and classical strong direct product theorems and optimal time-space tradeoffs. *SIAM J. Comput.*, 36(5):1472–1493, 2007.

21 S. Laplante and F. Magniez. Lower bounds for randomized and quantum query complexity using Kolmogorov arguments. *SIAM J. Comput.*, 38(1):46–62, 2008.

22 T. Lee, R. Mittal, B. W. Reichardt, R. Špalek, and M. Szegedy. Quantum query complexity of state conversion. In *Proc. FOCS'11*, pages 344–353, 2011.

23 T. Lee and J. Roland. A strong direct product theorem for quantum query complexity. In *Proc. CCC'12*, pages 236 – 246, 2012.

24 N. Nisan and M. Szegedy. On the degree of Boolean functions as real polynomials. *Comput. Complex.*, 4:301–313, 1994.

25 B. W. Reichardt. Reflections for quantum query algorithms. In *Proc. SODA'11*, pages 560–569, 2011.

26 B. W. Reichardt and R. Špalek. Span-program-based quantum algorithm for evaluating formulas. In *Proc. STOC'08*, pages 103–112, 2008.

27 A. A. Sherstov. Strong direct product theorems for quantum communication and query complexity. In *Proc. STOC'11*, pages 41–50, 2011.

28 R. Špalek. The multiplicative quantum adversary. In *Proc. CCC'08*, pages 237–248, 2008.

29 R. Špalek and M. Szegedy. All quantum adversary methods are equivalent. *Theor. Comput.*, 2:1–18, 2006.

30 S. Zhang. On the power of Ambainis lower bounds. *Theor. Comput. Sci.*, 339(2):241–256, 2005.